Optimization approach for clustering datasets with weights

R. GHOSH, A. RUBINOV* and J. ZHANG

School of Information Technology and Mathematical Sciences and Centre for Informatics and Applied Optimization, University of Ballarat, P.O. Box 663 Ballarat, Victoria 3353, Australia

We introduce datasets with weights and suggest using the minimization of some highly nonsmooth functions for clustering of such datasets. Datasets with weights often appear as the result of an approximation of large-scale datasets. We examine such approximations and also consider the application of datasets with weights to examine self-organizing maps. Results of some numerical experiments are presented and discussed.

Keywords: Datasets with weights; Nonsmooth optimization; Cluster function; Bradley–Mangasarian approximation; Skeleton; Self-organizing map

1. Introduction

The structure of a finite set of points in finite dimensional space is important for many applications. We can use different tools in order to define and describe this structure. Successful application of these tools depends on the structure of the set in hand. Currently, the unsupervised classification (clustering) is one of the main tools for the description of the structure.

The subject of the cluster analysis is the partition of a finite set A into a given number k of overlapping or disjoint subsets A^i, $i = 1, \ldots, k$ with respect to predefined criteria such that

$$A = \bigcup_{i=1}^{k} A^i.$$

The sets A^i, $i = 1, \ldots, k$ are called clusters.

An excellent up-to-date survey of existing approaches is provided in ref. [5] and a comprehensive list of literature on clustering algorithms is available in this paper. Recall the definition of clustering analysis given in ref. [5]:

> Cluster Analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than that of a petter belonging to a different cluster.

*Corresponding author. Email: a.rubinov@ballarat.edu.au

It follows from this definition that the notion of clustering is relatively *flexible.*

Usually, the following hypothesis is implicitly accepted: the number of clusters of the set under consideration is substantially less than the number of its points. Then the clustering can give some impression on the structure of the set. However, this hypothesis is not always true. For example, if a set A is a uniform grid, then the most natural set of clusters is this grid itself (each point $a \in A$ is a cluster). If we are looking for ball-shaped clusters, then the search for the clusters of a set A can be reduced for the search of centers of these clusters. A collection of these centers $(\bar{x}_1, \ldots, \bar{x}_k)$ can be found as a minimizer of the so-called cluster function. If a set A consists of flat pieces, we can use hyperplanes for approximation of this set. In this paper, we study the so-called Bradley–Mangasarian approximation by hyperplanes and also an approximation by k-skeletons.

Often we need to transform the dataset in hand in order to get a new dataset, which is more convenient for investigation, in particular for clustering. Different types of transformations can be used. The result of a transformation is a dataset B that is simpler than the original dataset A. In particular, many points from the original dataset can be stuck together, that is, to have the same image in a new transformed dataset. Thus, each point $b \in B$ has an indicator that shows how many points from A are represented by b. We shall call this indicator the weight of b. The dataset B such that each $b \in B$ has a weight will be called a dataset with weights. Of course, weights can be considered as a new attribute of the records $b \in B$. However, we demonstrate in this paper, that often weights play a special role in the clustering procedure, so we need to consider this attribute separately.

We describe two situations where datasets with weights can be used. One of them is an approximation of large-scale datasets and the other is a quantization by means of self-organizing maps (SOM).

The paper has the following structure. In section 2, we recall the definition of cluster function. Bradley–Mangasarian approximation and skeletons are discussed in section 3. Datasets with weights are introduced in section 4. We also provide necessary conditions for some kind of Bradley–Mangasarian approximations and skeletons for datasets with weights in this section. Approximation of large-scale datasets by datasets with weights is examined in section 5. Quantization by means of SOM that leads to datasets with weights is discussed in section 6. An experimental discussion can be found in section 7.

2. Ball-shaped clusters and cluster functions

Let $A \subset \mathbb{R}^n$ be a finite set. Assume that we are looking for ball-shaped clusters of A. Then the search for clusters can be reduced to the search of centers of clusters. We say that $\bar{X} = (\bar{x}^1, \ldots, \bar{x}^k)$ is a set of the centers of k clusters of A if $d(A, \bar{X}) \leq d(A, X)$ for each $X = (x^1, \ldots, x^k)$. Here,

$$d(A, X) = \sum_{a \in A} d(a, X) = \sum_{a \in A} \min_{i=1,\ldots,k} ||a - x_i||.$$

Assume that centers of clusters $(\bar{x}^1, \ldots, \bar{x}^k)$ are known. Then the cluster i consists of all points $a \in A$ such that $||x_i - a|| < \min_{j \neq i} ||x_j - a||$. (If the equality $||x_i - a|| = ||x_{i'} - a|| = \min_{j \neq i} ||x_i - a||$ holds, then we can consider a as a member of either cluster i or cluster i') The search for centers of clusters, hence, the search for the clusters themselves, can be reduced to the following unconstrained minimization problem:

$$\text{minimize } C_k(x^1, \ldots, x^k) \quad \text{subject to } (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \tag{1}$$

where

$$C_k(x^1, \ldots, x^k) = \frac{1}{m} \sum_{a \in A} \min_{s=1,\ldots,k} ||x^s - a||. \tag{2}$$

The function C_k defined by equation (2) is called the *cluster function* [1]. The problem (1) depends on the choice of a norm: different norms can lead to different centers of clusters. Since clustering is a flexible notion, the use of different norms is acceptable.

The number of variables in the optimization problem (1) is $k \times n$. If the number k of clusters and the number n of attributes are large, we have a large-scale optimization problem. Since the notion of cluster is flexible, it is enough to get a deep enough local minimum of the cluster function in order to have a satisfactory description of centers of clusters.

3. Clustering by means of hyperplanes

Assume that we are looking for clusters of a set A and this set consists of flat pieces. Bradley and Mangasarian [3] suggested to use hyperplanes instead of points (centers of clusters) for clustering in such a case.

Let $H = \{x\colon [l, x] = c\}$ be a hyperplane. Here, $[l, x] = \sum_i l_i x_i$ is the inner product of vectors l and x. Assume that $\mathbb{R}^n$ is equipped with a norm $||\cdot||$. Then the distance $d(x, H)$ from a point x to H is equal to $|[l/||l||_*, x] - c|$, where $||l||_* = \max_{||x||=1} [l, x]$ is the conjugate norm. Suppose that we wish to find k hyperplanes that approximate the set A with card $A = m$. It was suggested in ref. [3] to find a family hyperplanes $H_i = \{x\colon [l_i, x] = c_i\}$, $i = 1, \ldots, k$ that minimize the sum of squares of the 2-norm distances between each point and a nearest to this point hyperplane from the family, that is to solve the following optimization problem:

$$\text{minimize } \frac{1}{m} \sum_{a \in A} \min_{i=1,\ldots,k} ([l_i, a] - c_i)^2 \quad \text{subject to } ||l_i||_2 = 1, \quad i = 1, \ldots, k \tag{3}$$

Then the cluster i consists of all points $a \in A$ such that $|[l_i, a] - c_i]| < \min_{i \neq i} |[l_j, a] - c_i|$. We shall call a solution of problem (3) a Bradley–Mangasarian approximation of order k for the set A. The function

$$G_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{a \in A} \min_{i=1,\ldots,k} ([l_i, a] - c_i)^2, \quad l_j \in \mathbb{R}^n, c_j \in \mathbb{R}, j = 1, \ldots k, \tag{4}$$

will be called Bradley–Mangasarian function.

Consider a version of the discussed definition, where instead of squares of 2-norm distances, the distance itself with respect to a certain norm $||\cdot||$ is considered. In other words, consider the optimization problem

$$\text{minimize } L_k((l_1, c_1), \ldots, (l_k, c_k)) \quad \text{subject to } ||l_i||_* = 1, \quad i = 1, \ldots, k, \tag{5}$$

where

$$L_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{a \in A} \min_{i=1,\ldots,k} (|[l_i, a] - c_i|). \tag{6}$$

We shall call a solution of problem (5) a *k-skeleton* of a set A.

Example 3.1 Consider the set $A \subset \mathbb{R}^2$: $A = A' \cup (-A')$ with $A' = \{(1, q)\colon q = -2, \ldots, -1, 0, 1, \ldots, 2\}$ (figure 1).

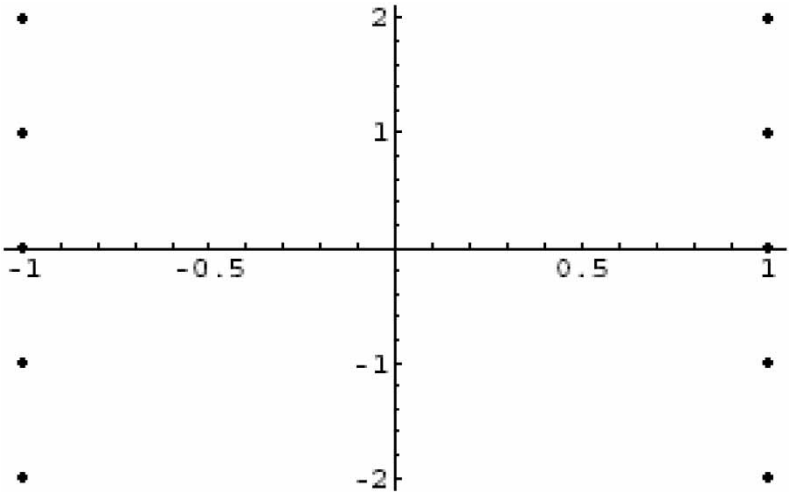

Figure 1. Dataset A.

Clearly, both the Bradley–Mangasarian approximation and the 2-skeleton of this set is the union of lines $\{(-1, x_2)\colon x_2 \in \mathbb{R}\}$ and $\{(1, x_2)\colon x_2 \in \mathbb{R}\}$. If $k = 3$, then an arbitrary approximation by straight lines consists of these two lines (one of them can appear twice). Now consider an approximation of A by one straight line. Using necessary conditions for minimum, presented in the next section (Proposition 4.4), one can show that problem (3) has only two local minimizers that define two straight lines. One of them is line $x_1 = 0$, the other is line $x_2 = 0$. The direct calculation shows that the solution of problem (3) with $k = 1$ is given by $l_1 = (1, 0)$, $c_1 = 0$, so the Bradley–Mangasarian approximation of this set is the line $x_1 = 0$. Consider now 1-skeleton for 2-norm.

It can be shown by direct calculation that this skeleton is the straight line $2x_1 + x_2 = 0$ which is going through points $(1, -2)$ and $(-1, 2)$. The symmetric line $2x_1 - x_2 = 0$ which is going through points $(-1, -2)$ and $(1, 2)$ is also a 1-skeleton. Each of these two lines is also 1-skeleton with respect to $||\cdot||_\infty$. (One can check it using necessary conditions that easily follows from Remark 4.1). Thus 1-skeleton coincides for two different norms. On the other hand, the Bradley–Mangasarian approximation and the skeleton for 2-norm are quite different.

4. Datasets with weights

Let $B \subset \mathbb{R}^n$ be a finite set. Assume that a positive number m_b is given for each $b \in B$. In such a case m_b is called a weight of b and B is called a *dataset with weights.* The simplest interpretation of a weight is as follows. Assume that each point $b \in B$ can be taken into account more than once. Then the weight m_b indicates how many times a point b appeared. Such an interpretation leads to the following definition of the generalized cluster function $\tilde{C}_k$ for a dataset B with weights:

$$\tilde{C}_k(x_1, \ldots, x_k) = \frac{1}{m} \sum_{b \in B} m_b \min_{i=1,\ldots,k} ||x_i - b||, \quad x_1, \ldots, x_k \in \mathbb{R}^n,$$

where $m = \sum_{b\in B} m_b$. An analog of the Bradley–Mangasarian function (4) has the following form:

$$\tilde{G}_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{b\in B} \min_{i=1,\ldots,k} m_b([l_i, b] - c_i)^2.$$

An analog of function L_k that serves for the definition of the k-skeleton has the following form:

$$\tilde{L}_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{b\in B} m_b \min_{i=1,\ldots,k} (|[l_i, b] - c_i|). \tag{7}$$

We now provide a theoretical analysis of optimization problems that related to the search for the 1-skeleton and the Bradley–Mangasarian approximation of order 1 for datasets with weights.

Consider an optimization problem

$$\text{minimize } f(x) \quad \text{subject to } x \in \Omega, \tag{8}$$

where Ω is a set and f is a DC function. The latter means that f can be represented in the form $f = f_1 - f_2$, where f_1 and f_2 are finite convex functions defined on $\mathbb{R}^n$. We need the cone $\Gamma(x, \Omega)$ of feasible elements at a point $x \in \Omega$ to the set Ω in order to describe the necessary conditions for a minimum. Recall that $u \in \Gamma(x)$ if there exist sequences $u_j \to u$ and $\alpha_j \to 0$ such that $x + \alpha_j u_j \in \Omega$. Let K be a convex cone. The conjugate to K cone $\{l : [l, x] \geq 0$ for all $x \in K\}$ will be denoted by K^*. Subdifferential of a convex function g at a point x will be denoted by $\partial g(x)$.

PROPOSITION 4.1 *Let x be a local solution of problem (8). Assume that the cone $\Gamma(x, \Omega)$ is convex. Then*

$$\partial f_1(x) - \Gamma^*(x, \Omega) \supset \partial f_2(x). \tag{9}$$

A proof of this well-known proposition can be found for example in ref. [4, Theorem 16.3].

Remark 4.1 If f is convex, that is $f_1 = f$ and $f_2 = 0$, then necessary conditions (9) have the form $0 \in \partial f(x) - \Gamma^*$, which is equivalent to

$$\Gamma^* \cap \partial f(x) \neq \emptyset. \tag{10}$$

We now give necessary conditions for the 1-skeleton and the Bradley–Mangasarian approximation of the order 1 for datasets with weights. Let B be a dataset with weights $(m_b)_{b\in B}$ and $m = \sum_{b\in B} m_b$. Consider a function L_1 that serves for the determining a 1-skeleton. Then

$$\hat{L}_1 := m\tilde{L}_1(l,\ c) = \sum_{b\in B} m_b|[l, b] - c|. \tag{11}$$

Function $\hat{L}_1$ is convex. We now calculate $\partial \hat{L}_1(l, c)$. For $(l, c) \in \mathbb{R}^{n+1}$, set

$$B^+(l, c) = \{b \in B : [l, b] - c > 0\}, \quad B^-(l, c) = \{b \in B : [l, b] - c < 0\}, \tag{12}$$

$$B^0(l, b) = \{b \in B : [l, b] - c = 0\}. \tag{13}$$

PROPOSITION 4.2 *Let $(l, c) \in \mathbb{R}^{n+1}$. Then $(u, v) \in \partial \hat{L}_1(l, c)$ if and only if for each $b \in B^0(l, c)$ there exists $\alpha^b \in [0, 1]$ such that*

$$u = \sum_{b \in B^+(l,c)} m_b b - \sum_{b \in B^-(l,c)} m_b b + \sum_{b \in B^0(l,c)} m_b (2\alpha^b - 1) b,$$

$$v = -\sum_{b \in B^+(l,c)} m_b + \sum_{b \in B^-(l,c)} m_b + \sum_{b \in B^0(l,c)} m_b (1 - 2\alpha^b)$$

Proof For a vector $b \in \mathbb{R}^n$, define

$$s_b(l, c) = |[l, b] - c| = \max([l, b] - c, -[l, b] + c). \tag{14}$$

Using the subdifferential calculus, we get

$$\partial s_b(l, c) = \begin{cases} \{(b, -1)\}, & [l, b] - c > 0 \\ \{(-b, 1)\}, & [l, b] - c < 0 \\ \{(2\alpha - 1)b, 1 - 2\alpha) \colon \alpha \in [0, 1]\} & [l, b] - c = 0 \end{cases} \tag{15}$$

We have

$$\hat{L}_1(l, c) = \sum_{b \in B} m_b s_b(l, c) = \sum_{b \in B^+(l,c)} m_b s_b(l, c) + \sum_{b \in B^-(l,c)} m_b s_b(l, c) + \sum_{b \in B^0(l,c)} m_b s_b(l, c).$$

It follows from equation (15) that $(u, v) \in \partial \hat{L}_1(l, c)$ if and only if for each $b \in A^0(l, c)$ there exists $\alpha^b \in [0, 1]$ such that

$$(u, v) = \left(\sum_{b \in B^+(l,c)} m_b (b, -1) + \sum_{b \in B^-(l,c)} m_b (-b, 1) + \sum_{b \in B^0(l,c)} m_b ((2\alpha^b - 1)b, 1 - 2\alpha^b)) \right).$$

It means that

$$u = \sum_{b \in B^+(l,c)} m_b b - \sum_{b \in B^-(l,c)} m_b b + \sum_{b \in B^0(l,c)} m_b (2\alpha^b - 1) b \tag{16}$$

$$v = -\sum_{b \in B^+(l,c)} m_b + \sum_{b \in B^-(l,c)} m_b + \sum_{b \in B^0(l,c)} m_b (1 - 2\alpha^b) \tag{17}$$

Thus, the result follows. ■

Let $\hat{G}_1 = m\tilde{G}_1(l, c)$, where $\tilde{G}_1$ is the generalized Bradley–Mangasarian function of order 1. Then

$$\hat{G}_1(l, c) = \sum_{b \in B} m_b ([l, b] - c)^2$$

is a convex function. This function is differentiable and

$$\nabla \hat{G}_1(l, c) = \sum_{b \in B} 2m_b ([l, b] - c)(b, -1). \tag{18}$$

Assume that $\mathbb{R}^n$ is equipped with the norm $||\cdot||_2$ and let $S = \{x \colon ||x||_2 = 1\}$ be the unit sphere. It is easy to check (and well known) that

$$\Gamma(l, S) = \{u \colon [l, u] = 0\}. \tag{19}$$

Consider now the set $S \times \mathbb{R} \subset \mathbb{R}^{n+1}$. Let (l, λ) belongs to this set. It follows directly from the definition of the cone of feasible elements and equation (19) that

$$\Gamma \equiv \Gamma((l, \lambda), S \times \mathbb{R}) = \{(u, v) \colon [l, u] = 0, v \in \mathbb{R}\}. \tag{20}$$

Then

$$\Gamma^* = \{(\lambda l, 0) \colon \lambda \in \mathbb{R}\} \tag{21}$$

PROPOSITION 4.3 *Let (l, c) define a 1-skeleton H of a dataset B with weights $(m_b)_{b \in B}$ with respect to $||\cdot||_2$. Then there exists $\lambda \in \mathbb{R}$ and for each $b \in B^0$ there exists $\alpha^b \in [0, 1]$ such that*

$$\sum_{b \in B^+(l,c)} m_b b - \sum_{b \in B^-(l,c)} m_b b + \sum_{b \in B^0(l,c)} m_b (2\alpha^b - 1) b = \lambda l, \tag{22}$$

$$-\sum_{b \in B^+(l,c)} m_b + \sum_{b \in B^-(l,c)} m_b + \sum_{b \in B^0(l,c)} m_b (1 - 2\alpha^b) = 0. \tag{23}$$

The proof follows directly from Remark 4.4, Proposition 4.2 and equation (21). Applying equation (18) instead of Proposition 4.2, we conclude that the following assertion holds.

PROPOSITION 4.4 *Let (l, c) define a Bradley–Mangasarian approximation H of order 1 for a dataset B with weights $(m_b)_{b \in B}$. Then there exists $\lambda \in \mathbb{R}$ such that*

$$\sum_{b \in B} m_b ([l, b] - c) b = \lambda l; \quad \sum_{b \in B} m_b ([l, b] - c) = 0. \tag{24}$$

We can present equation (24) in the following form:

$$\sum_{b \in B^+(l,c)} m_b ([l, b] - c) b + \sum_{b \in B^-(l,c)} m_b ([l, b] - c) b = \lambda l \tag{25}$$

$$\sum_{b \in B^+(l,c)} m_b ([l,\ b] - c) + \sum_{b \in B^-(l,c)} m_b ([l, b] - c) = 0. \tag{26}$$

Let $d(b, H)$ be the distance between a point b and the hyperplane $H = \{x : [l, x] = c\}$. Then equations (25) and (26) can be rewritten as

$$\sum_{b \in B^+(l,c)} m_b d(b, H) b - \sum_{b \in B^-(l,c)} m_b d(b, H) b = \lambda l \tag{27}$$

$$\sum_{b \in B^+(l,c)} m_b d(b, H) - \sum_{b \in B^-(l,c)} m_b d(b, H) = 0 \tag{28}$$

Let us compare the necessary conditions for 1-skeleton H_{sk} of the set B, given by equations (22) and (23) and the necessary conditions for Bradley–Mangasarian approximation $H_{\mathrm{B-M}}$ of order 1, given by equations (27) and (28). We can conclude the following:

1) necessary conditions (22) and (23) depend on points $b \in B$ that are placed on the plane H_{sk} (i.e, on points belonging to $B^0(l, c)$). Necessary conditions (27) and (28) do not depend on these points;
2) the distances from points $b \in B$ to $H_{\mathrm{B-M}}$ are taken into account; the distances from $b \in B$ to H_{sk} do not play any role.

This means that the skeletons are quite different from the Bradley–Mangasarian approximations.

Proposition 4.1 can also be used for examination of k-skeletons and the Bradley–Mangasarian approximation of order k with $k > 1$. We demonstrate it for a Bradley–Mangasarian approximation of the order 2. The function

$$\tilde{G}_2((l_1, c_1), (l_2, c_2)) = \sum_{b \in B} m_b \min(([l_1, b] - c_1)^2, ([l_2, b] - c_2)^2)$$

is quasi-differentiable [4]. Using results of quasi-differentiable calculus, we can represent this function in the form $\tilde{G}_2 = f_1 - f_2$, where

$$f_1((l_1, c_1), (l_2, c_2)) = \sum_{b \in B} m_b(([l_1, b] - c_1)^2 + ([l_2, b] - c_2)^2),$$

$$f_2((l_1, c_1), (l_2, c_2) = \sum_{b \in B} m_b(\max(([l_1, b] - c_1)^2, ([l_2, b] - c_2)^2).$$

Both f_1 and f_2 are convex functions, function f_1 is differentiable and

$$\nabla f_1((l_1, c_1), (l_2, c_2)) = \sum_{b \in B} 2m_b(([l_1, b] - c_1)(b, -1) + ([l_2, b] - c_2)(b, -1)).$$

We now calculate the subdifferential ∂f_2 of f_2. Let

$$B_1 = \{b \in B \colon ([l_1, b] - c_1])^2 > [l_2, b] - c_2])^2\},$$

$$B_2 = \{b \in B \colon [l_2, b] - c_2])^2 > [l_1, b] - c_1])^2\}$$

$$B_3 = \{b \in B \colon [l_1, b] - c_1])^2 = [l_2, b] - c_2])^2\}.$$

Then

$$\begin{aligned}\partial f_2((l_1, c_1), (l_2, c_2)) = &\sum_{b \in B_1} 2m_b([l_1, b] - c_1)(b, -1) + \sum_{b \in B_2} 2m_b[l_2, b] - c_2])(b, -1) \\ &+ \sum_{b \in B_3} 2m_b\{\alpha_b([l_1, b] - c_1]) + (1 - \alpha_b)([l_2, b] - c_2])(b, -1) \colon \\ &\alpha_b \in [0, 1]\}.\end{aligned}$$

Using these expressions, Proposition 4.1 and equation (21), we can present the necessary conditions for a Bradley–Mangasarian approximation of the order 2.

5. Approximation of large-scale datasets

Datasets with weights often appears as an approximation a large-scale dataset. Large-scale datasets usually contain many points located in a bounded set. Thus, many points from this dataset are very close to each other. Let $A \subset \mathbb{R}^n$ be a finite set. Assume that a certain small neighborhood of a point $b \in \mathbb{R}^n$ contains m_b points from A. We can approximate each of these points by b and replace the corresponding part of the cluster function by one term $m_b \min_i ||x_i - b||$.

To be more precise, for a given A and for a given tolerance, ε consider a set $B \subset \mathbb{R}^n$, such that for each $a \in A$ there exists $b \in B$ with the property $||a - b|| < \varepsilon$. We say that a collection $(A_b)_{b\in B}$ of subsets of A is an ε-disjoint cover of A if

$$||a - b|| < \varepsilon (a \in A_b), \quad A_b \cap A_{b'} = \emptyset (b \neq b'), \quad A = \bigcup_{b\in B} A_b.$$

Let m be the cardinality of A and m_b be the cardinality of A_b. Clearly, $m = \sum_{b\in B} m_b$. Replacing each $a \in A_b$ with b in the presentation of the cluster function C_k, we obtained the following function

$$\tilde{C}_k(x_1, \ldots, x_k) = \frac{1}{m} \sum_{b\in B} m_b \min(||x_1 - b||, \ldots, ||x_k - b||),$$

which is the generalized cluster function corresponding to B. Note that m_b is a weight of a point b in the dataset B.

The following assertion demonstrates that generalized cluster functions can be used for approximation of cluster functions.

PROPOSITION 5.1 *Let $(A_b)_{b\in B}$ be an ε-disjoint cover of A and $\tilde{C}_k$ be the generalized cluster function corresponding to this cover. Then*

$$|C_k(x_1, \ldots, x_k) - \tilde{C}_k(x_1, \ldots, x_k)| < \varepsilon \quad \textit{for all } (x_1, \ldots, x_k) \in (\mathbb{R}^n)^k.$$

Proof We have

$$C_k(x_1, \ldots, x_k) = \frac{1}{m} \sum_{a\in A} \min_{i\leq k} ||x_i - a|| = \frac{1}{m} \sum_{b\subset B} \sum_{a\in A_b} \min_{i\leq k} ||x_i - a||.$$

As $||x_i - a|| \leq ||x_i - b|| + \varepsilon$ for all $i \leq k$ and $a \in A_b$, it follows that $\min_{i\leq k} ||x_i - a|| \leq \min_{i\leq k} ||x_i - b|| + \varepsilon$. Hence,

$$\sum_{a\in A_b} \min_{i\leq k} ||x_i - a|| \leq \sum_{a\in A_b} (\min_{i\leq k} ||x_i - b|| + \varepsilon) = m_b \min_{i\leq k} ||x_i - b|| + m_b\varepsilon),$$

where m_b is the cardinality of A_b. The same argument shows that

$$m_b \min_{i\leq k} ||x_i - b|| \leq \sum_{a\in A_b} \min_{i\leq k} ||x_i - a|| + m_b\varepsilon.$$

Hence,

$$\left| \sum_{a\in A_b} \min_{i\leq k} ||x_i - a|| - m_b \min_{i\leq k} ||x_i - b|| \right| < m_b\varepsilon.$$

Let $u = |C_k(x_1, \ldots, x_k) - \tilde{C}_k(x_1, \ldots, x_k)|$. As $\sum_{b\in B} m_b = m$, we have

$$u \leq \frac{1}{m} \left(\sum_{b\in B} \left| \sum_{a\in A_b} \min_{i\leq k} ||x_i - a|| - m_b \min_{i\leq k} (||x_i - b||) \right| \right) \leq \frac{1}{m} \sum_{b\in B} m_b\varepsilon = \varepsilon. \qquad \blacksquare$$

It follows from Proposition 5.1 that the approximating set B can be used for the search for centers of clusters of the set A with the tolerance ε. The search can be accomplished by

means of cluster function $\tilde{C}_k$. As this function contains weights m_b of each point $b \in B$, it is convenient to consider B as a dataset with weights.

A similar approach can be used for an approximate search for skeletons. Consider again a dataset B with weights such that collection $(A_b)_{b\in B}$ forms an ε-disjoint cover of A. Let m_b be the weight of b (the cardinality of A_b) and $m = \sum_{b\in B}$ be the cardinality of A. Replacing each $a \in A_b$ with b, we shall have the following function

$$\tilde{L}_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{b\in B} m_b \min_{i=1,\ldots,k} |[l_i, b] - c_i]| \tag{29}$$

instead of function $L_k(l_1, c_1), \ldots, (l_k, c_k))$ defined by equation (6).

PROPOSITION 5.2 *Let $(A_b)_{b\in B}$ be an ε-disjoint cover of A and $\tilde{L}_k$ be function defined by equation* (29)*. Then for all $((l_1, c_1), \ldots, (l_k, c_k)) \in (\mathbb{R}^n \times \mathbb{R})^k$, we have*

$$|L_k((l_1, c_1), \ldots, (l_k, c_k)) - \tilde{L}_k((l_1, c_1), \ldots, (l_k, c_k)| < \varepsilon \tag{30}$$

Proof We have

$$L_k((l_1, c_1), \ldots, (l_k, c_k)) = \frac{1}{m} \sum_{a\in A} \min_{i\le k} |[l_i, a] - c_i| = \frac{1}{m} \sum_{b\in B} \sum_{a\in A_b} \min_{i\le k} |[l_i, a] - c_i|. \tag{31}$$

Let $i = 1, \ldots, k$. As $||a - b|| < \varepsilon$ for $a \in A_b$ and $||l||_* = 1$, it follows that

$$|[l_i, a] - c_i| \le |[l_i, b] - c_i| + |[l_i, a - b]| \le |[l_i, b] - c_i| + \varepsilon. \tag{32}$$

Hence,

$$\min_i |[l_i, a] - c_i| \le \min_i |[l_i, b] - c_i| + \varepsilon, \quad a \in A_b$$

and

$$\sum_{a\in A_b} \min_i |[l_i, a] - c_i| \le m_b \min_i |[l_i, b] - c_i| + m_b\varepsilon.$$

A similar argument shows that

$$m_b \min_i |[l_i, b] - c_i| \le \sum_{a\in A_b} \min_i |[l_i, a] - c_i| + m_b\varepsilon.$$

Hence,

$$\left| \sum_{a\in A_b} \min_{i=1,\ldots,k} |[l_i, a] - c_i| - m_b \min_{i=1,\ldots,k} |[l_i, b] - c_i| \right| \le m_b\varepsilon. \tag{33}$$

The same argument as in the proof of Proposition 5.1 shows that equation (33) implies equation (30). ■

As the notion of clusters is flexible, we can use a uniform approximation of the cluster function for the search of ball-shaped clusters and a uniform approximation of function L_k for the search of flat clusters. Propositions 5.1 and 5.2 show that these approximation can be constructed by using datasets with weights that appear as an approximation of a given dataset.

6. Quantization using a SOM

SOM is another form of clustering technique which leads to datasets with weights. This approach is based on the concept of vector quantization with competitive learning. SOM (figure 2) was developed by Kohonen [6]. It is an unsupervised learning algorithm. It creates a relationship among the vectors from a given dataset (input), which is based on a certain set (discrete lattice) B on the plane. This set is usually called a map.

SOM is a special kind of unsupervised neural network that projects high-dimensional data vectors into two-dimensional plane [see, for example, 7]. The basic motivation behind this is to cluster dataset in a low-dimensional space. Two-dimensional maps are also useful for visualization of a high dimensional system. There are two phases involved in creating SOM. In the first phase, the original data set A is trained and the connection weights from input layer to the individual nodes in the map B are obtained; see ref. [8] for details. After the training, each input vector a is mapped into one of possible points b in the grid B using the minimum Euclidian distance through its weights. A detailed description of algorithms for training and determining weights can be found in ref. [6].

However, there are two major problems with the SOM. SOM model requires a predefined map structure, before the training of its weight. Finding a proper map size is based on trial and error. To overcome this problem, training of SOM can be performed with sufficiently large map size. Experimental studies also suggest that having the map size much larger (compared with the clusters) produces better results than that with a smaller map size (compared with the cluster). In that case, the major problems are that the complete learning process has to be repeated for different map sizes, if the size of the map is very small, then the classification error for every input can be very high, resulting dissimilar vectors being assigned to same point or similar vector can be assigned to different points belonging to the map. One solution to this problem, which are adapted by researchers is to consider a sufficiently large map size for training. Hence, let us assume that we want to make k cluster, so the map size should be $N \times N$, where $N \gg k$. We can then find the weight (that is also called *confidence* in this

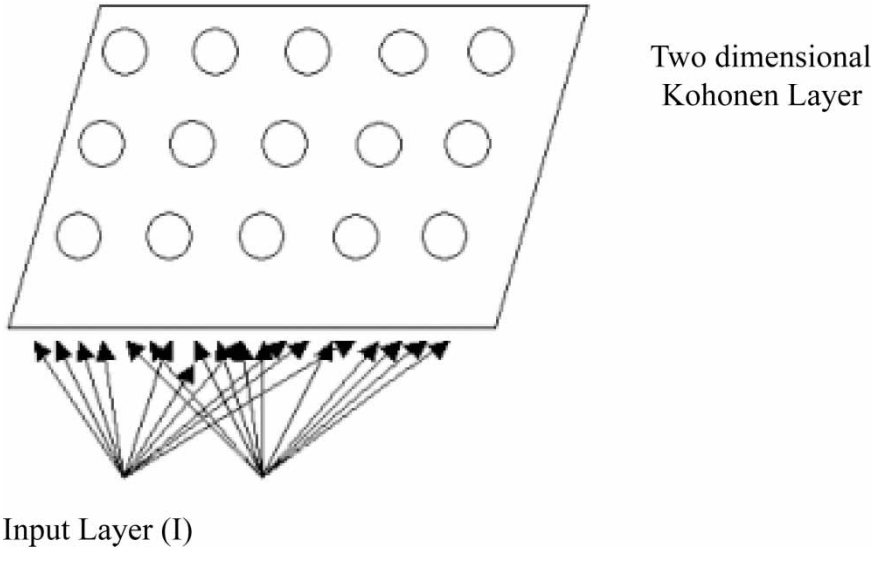

Figure 2. First abstraction level using Kohonen map.

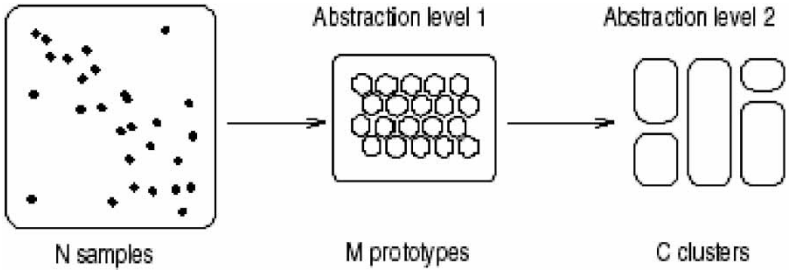

Figure 3. Two levels of abstraction.

theory) of every points belonging to the map by calculating the frequency of records from the input dataset A, that are projected into the same points. Finally, we get a set where the cardinal value is three, where the first two represents the coordinates of points belonging to the map and the third one representing weights of that particular point. It is convenient to consider this set as a dataset (map) with weights. This serve a twofold solution: firstly, the problem of finding clusters using optimization technique applied to the original high-dimensional dataset can be reduced to a two-dimensional problem. Finally, further clustering the points on the map reduces the sensitivity problem for the SOM with the map size. In addition, the curse of many cluster points in the SOM could be solved. Thus, we can say that SOM could be a useful tool for using clustering for the initial abstraction level to form some prototypes for the clustering.

Figure 3 illustrates the two levels of abstraction from the original dataset. In this figure, we can see that the original dataset A is first transformed into two-dimensional SOM B on the plane to form M clusters prototypes. From these prototypes, we compute the weight (confidence) of each point according to the frequency of the original data that have been mapped into a cluster prototype, and then finally the optimization technique is used to find k different clusters. At this point, we assume that the sensitivity of the map size will be captured by the confidence of the points; hence, within a sufficiently large range of map size the second level of abstraction will be unaffected.

Because a map B obtained as the result of SOM is a dataset with weights, we can use a minimization of generalized cluster function for finding centers of clusters of this dataset.

As B is a two-dimensional dataset, the dimension of corresponding minimization problem is $2k$, where k is the number of clusters. If k is not very large, we get an optimization problem that can be solved by modern methods of global optimization. If the number of clusters is bigger, we can use hierarchical clustering. We also can use generalized Bradley–Mangasarian function for finding straight lines that approximate B in the sense of Bradley–Mangasarian and use function $\tilde{L}$ defined by equation (7) in order to find skeletons of this set. In both cases, we again have an optimization problem, whose dimension is much less than the dimension of a problem that is used for clustering the original dataset.

7. Experimental discussion

We applied the approach proposed in section 6 for *credit screening dataset*, which is one of the benchmark datasets from UCI machine learning repository. The description of the database is given as follows.

Table 1. Map B with weights.

Line	Point/weight	Point/weight	Point/weight	Point/weight	Point/weight
0	(0, 0)/67	(0, 1)/57	(0, 2)/141	(0, 3)/36	(0, 4)/138
1	(1, 0)/10	(1, 1)/24	(1, 2)/24	(1, 3)/30	(1, 4)/32
2	(2, 0)/10	(2, 1)/17	(2, 2)/17	(2, 3)/14	(2, 4)/13
3	(3, 0)/11	(3, 1)/1	(3, 2)/5	(3, 3)/6	(3, 4)/4
4	(4, 0)/9	(4, 1)/11	(4, 2)/8	(4, 3)/1	(4, 4)/5

Credit screening dataset: The source of this dataset has not been made public. This dataset was submitted by Quinlan and was used earlier and published in ref. [9]. The total number of instances is 690 and there are 15 attributes in total. There are 37 missing values, which was replaced by mean of the corresponding input variables. six out of 15 variables are continuous and the remaining variables are discrete. The dataset can be obtained online from UCI machine learning repository and the ftp address is ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/.

We considered different types of SOM and use the algorithm from ref. [7] that was mentioned in section 6 for development SOM. Here, we presented some results only for 5×5 map size. This map B consists of 25 points that are uniformly distributed in the square $[0, 4] \times [0, 4]$ on the plane (table 1).

Each of these points has a nonzero weight. If we consider map B without weights, we shall have a uniform grid consisting of 25 points. Each of these points can be considered as an independent cluster. In such a case, we shall have many clusters. To reduce their number, we can consider B as a dataset with weights and find clusters in B. The graph depicted in figure 4 shows the SOM with weights for the dataset B. The following observation is important: the distribution of weights is not uniform (table 1 and figure 4). Many 'heaviest' points are close to one side of the square B. (Compare weights for points from different lines in table 1.)

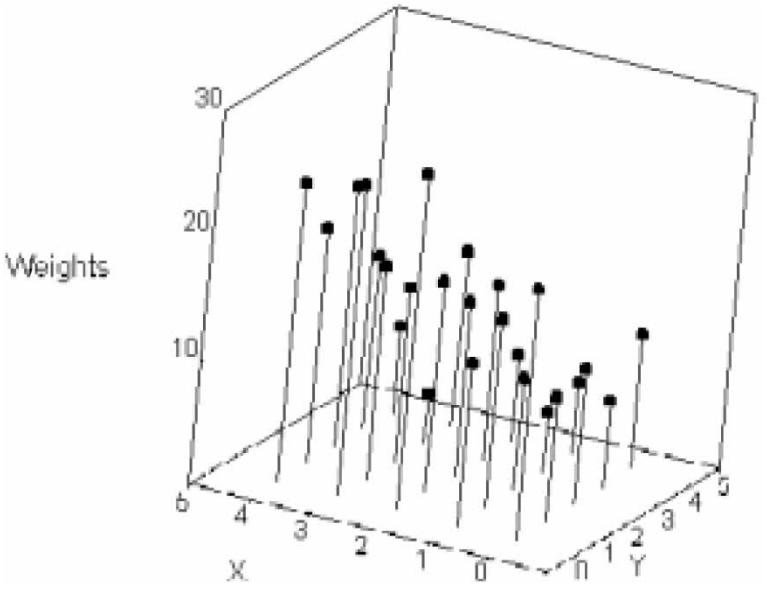

Figure 4. Weights of points from initial abstraction level.

Table 2. Centers of clusters.

Norm	Number of clusters	Cluster	x_1	x_2
1	1	1	0.00	2.00
	2	1	2.00	2.00
		2	0.00	3.00
	3	1	0.00	1.99
		2	2.00	2.00
		3	1.00	2.00
	4	1	1.99	2.90
		2	0.00	1.99
		3	0.99	1.99
		4	3.99	1.00
2	1	1	0.28	2.13
	2	1	0.00	3.99
		2	0.35	1.40
	3	1	0.09	1.63
		2	0.00	3.99
		3	2.66	1.69
	4	1	0.00	3.99
		2	0.00	2.00
		3	2.76	1.90
		4	0.163	0.40

We use the minimization of generalized cluster function $\tilde{C}_k$ for finding centers of clusters and also the minimization of Bradley–Mangasarian function $\tilde{G}_k$ and function $\tilde{L}_k$ for funding Bradley–Mangasarian approximation and skeletons, respectively. We construct functions $\tilde{C}_k$ using norms $||\cdot||_1$ and $||\cdot||_2$. We consider $\tilde{L}_k$ with respect to norms $||\cdot||_2$ and $||\cdot||_\infty$. (Note $(||\cdot||_\infty)_* = ||\cdot||_1$.) For the minimization, we use the numerical method for global optimization that was described in ref. [2]. This is a hybrid between local discrete gradient method and global simulation annealing method.

Table 2 shows the centers for credit screening data base with map size 5×5, with different norms and various number of clusters. Table 3 shows straight lines that are either skeletons or

Table 3. Approximation by straight lines.

Norm	Type of approximation	Number of lines	Equations of lines
∞	Skeleton	1	$-0.75x_1 + 0.24x_2 = -20.23$
		2	$x_1 = 0$
			$0.63x_1 + 0.37x_2 = -2.11$
		3	$0.36x_1 + 0.64x_2 = -1.54$
			$0.97x_1 + 0.03x_2 = -0.10$
			$0.40x_1 - 0.60x_2 = 1.45$
2	Skeleton	1	$0.10x_1 + 0.0003x_2 = -0.0003$
		2	$0.10x_1 + 0.02x_2 = -0.05$
			$0.77x_1 + 0.64x_2 = -3.34$
		3	$0.10x_1 + 0.07x_2 = -1.21$
			$0.10x_1 + 0.002x_2 = -0.006$
			$0.88x_1 + 0.47x_2 = -3.55$
	Bradley–Mangasarian approximation	1	$0.97x_1 + 0.23x_2 = -1.18$
		2	$0.81x_1 + 0.60x_2 = -3.27$
			$0.99x_1 + 0.12x_2 = -0.42$
		3	$0.76x_1 - 0.64x_2 = -2.06$
			$0.10x_1 - 0.07x_2 = 0.05$
			$-0.84x_1 + 0.54x_2 = 0.28$

carry out Bradley–Mangasarian approximation. These tables demonstrate that the clustering significantly depends on the choice of norm. The question which norm is more appropriate is open. It is interesting to find some classes of datasets for which $||\cdot||_1$ is more preferable than $||\cdot||_2$ and vice versa. We can also conclude that centers of clusters are displaced in the direction of heaviest points. Straight lines that are used for description of clusters are also displaced in this direction. Skeletons and Bradley–Mangasarian approximations are presented in figures 5 and 6 respectively.

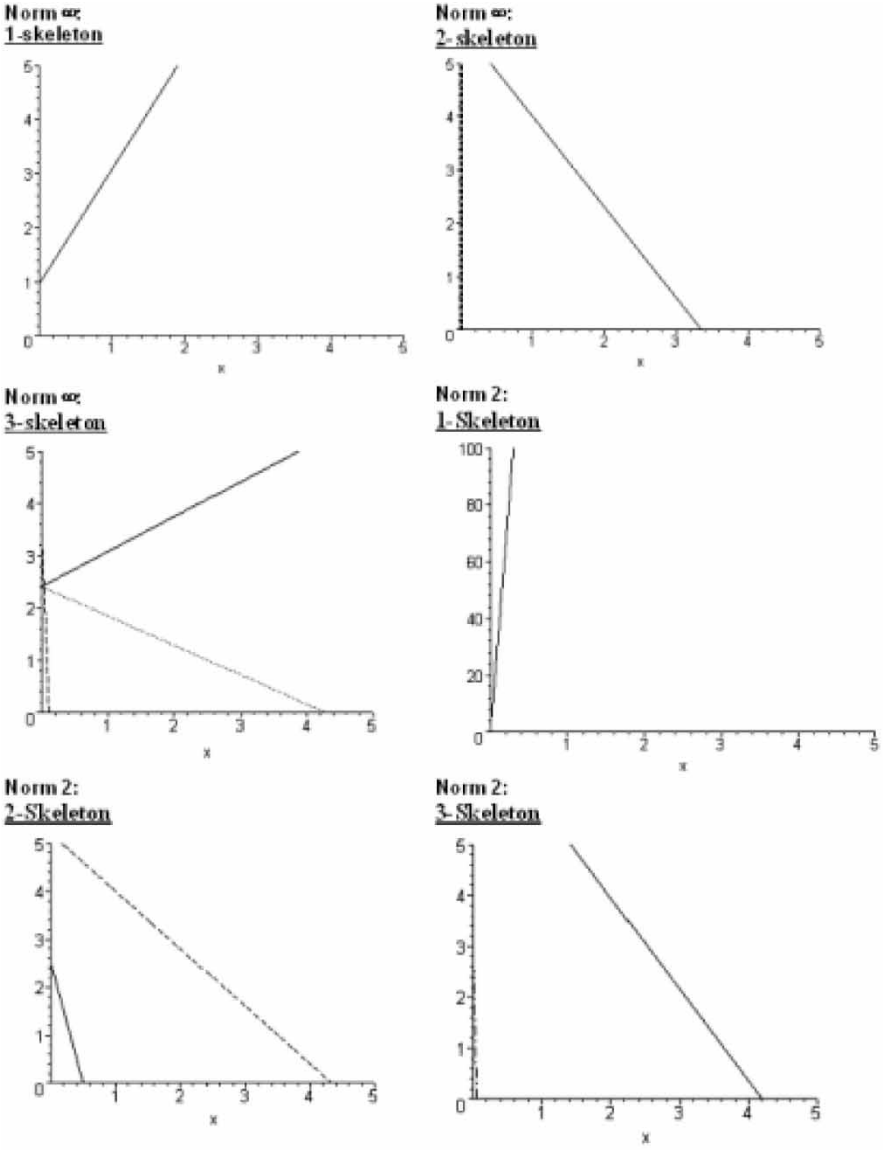

Figure 5. Skeletons.

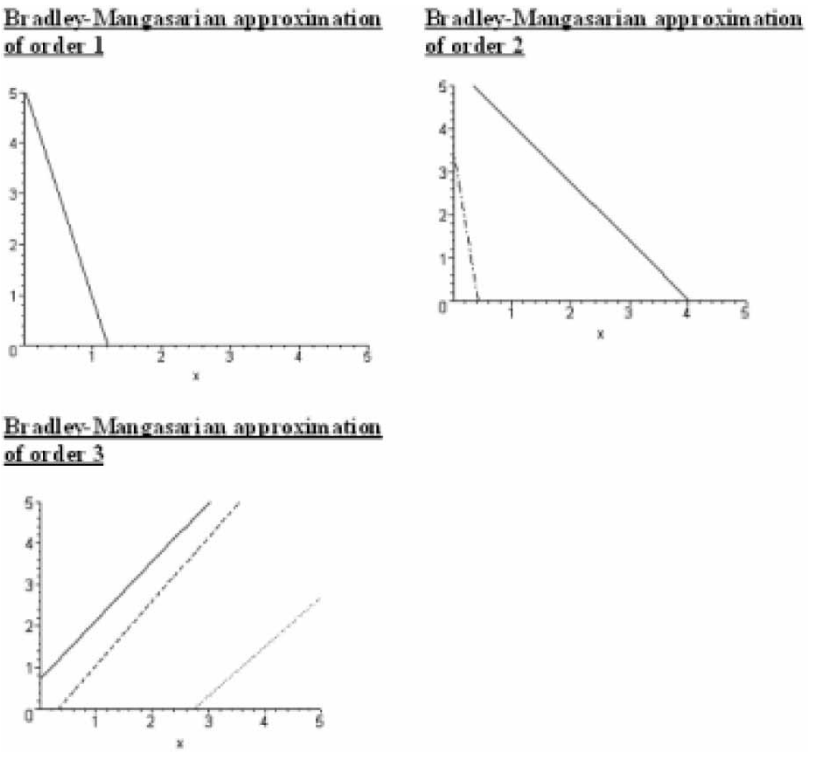

Figure 6. Bradley–Mangasarian approximations.

8. Conclusions

1. In this paper, we introduce the notion of a dataset with weights and demonstrate that such a dataset can appear as the result of approximation of a large-scale dataset and as the result of a quantization by a SOM.
2. We suggest to use the minimization of a generalized cluster function for the search of ball-shaped clusters in datasets with weights.
3. We discuss possible application of hyperplanes for the search of clusters in datasets with weights. We consider the notion of a Bradley–Mangasarian approximation and the notion of a skeleton and compare them. We present the necessary conditions for the Bradley–Mangasarian approximation of the first order and for the 1-skeleton. Using these conditions, we demonstrate that these two kinds approximation often lead to quite different results.
4. We show that an approximation of a large-scale dataset A by means of a small dataset B with weights leads to the uniform approximation of the cluster function for A by the generalized cluster function for B. The similar result holds for functions that serve for the search of skeletons.
5. We provide an example that show that approximation of a dataset with weights given by either centres of clusters or by collections of hyperplanes heavily depends on the choice of norm. The determining norm which is good for clustering of a given dataset is an open question.

Acknowledgement

The authors are thankful to an anonymous referee for their valuable comments.

References

[1] Bagirov, A.M., Rubinov, A.M. and Yearwood, J., 2002, *Optimization and Engineering*, **3**, 129–155.
[2] Bagirov, A.M. and Zhang, J., 2003, *Proceedings of Industrial Mathematics Symposium* (Perth).
[3] Bradley, P.S. and Mangasarian, O.L., 2000, *Journal of Global Optimization*, **16**, 23–32.
[4] Demyanov, V.F. and Rubinov, A.M., 1986, *Quasidifferential Calculus* (NY: Optimization Software, Inc.).
[5] Jain, A.K., Murty, M.N. and Flynn, P.J., 1999, *ACM Computing Surveys*, **31**, 264–323.
[6] Kohonen, T., 1982, *Biological Cybernetics*, **43**, 59–69.
[7] Lampimen, J. and Oja, E., 1992, *Journal of Mathematical Imaging and Vision*, **2**, 261–272.
[8] Mangiameli, P., Chen, S.K. and West, D., 1996, *European Journal of Operation Research*, **93**(2).
[9] Quinlan, J.R., 1987, *International Journal of Man-Machine Studies*, **27**, 221–234.